

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

# Segmentace obrazu s využitím metod založených na grafových řezech

## Image Segmentation using Methods based on Graph Cuts

## Zadání bakalářské práce

Student:

**Michal Maier**

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Segmentace obrazu s využitím metod založených na grafových řezech  
Image Segmentation using Methods based on Graph Cuts

Jazyk vypracování:

čeština

Zásady pro vypracování:

Segmentace obrazu je technika počítačového vidění umožňující rozdělení vstupního obrazu do dílčích částí. Jednou skupinou segmentačních algoritmů jsou metody založené na grafových řezech, ať už v původní podobě nebo v metodách z grafových řezů vycházejících. Cílem této práce je seznámit se s metodami založenými na grafových řezech (např. Graph Cuts, Grabcut, Onecut) a porovnat jejich segmentační výsledky na množině reálných obrazů.

Ve své práci proveďte:

1. Nastudujte a popište vybrané segmentační metody založené na grafových řezech.
2. S využitím dostupných knihoven vytvořte aplikaci pro segmentaci obrazu pomocí vybraných metod.
3. Otestujte metody na reálných obrazech a porovnejte jejich výsledky.
4. Zhodnoťte dosažené výsledky.

Seznam doporučené odborné literatury:

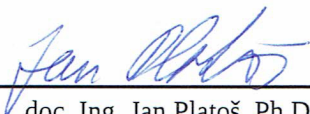
- [1] Y. Boykov and M.-P. Jolly. "Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images." In IEEE International Conference on Computer Vision, pp. 105–112 vol.1, 2001
- [2] C. Rother, V. Kolmogorov, A. Blake. "Grabcut - interactive foreground extraction using iterated graph cuts." In ACM transactions on Graphics (SIGGRAPH), pp. 309-314, 2004
- [3] M. Tang, L. Gorelick, O. Veksler, Y. Boykov. "GrabCut in One Cut." In IEEE International Conference on Computer Vision, Sydney, VIC, pp. 1769-1776, 2013

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **Ing. Michael Holuša, Ph.D.**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2020

  
\_\_\_\_\_  
doc. Ing. Jan Platoš, Ph.D.  
vedoucí katedry



  
\_\_\_\_\_  
prof. Ing. Pavel Brandštetter, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 15. května 2020

  
.....

Rád bych na tomto místě poděkoval vedoucímu této práce Ing. Michaelu Holušovi, Ph.D. za příkladné vedení a cenné rady v průběhu tvorby této práce, protože by tato práce bez jeho pomoci nevznikla.

## **Abstrakt**

Tato bakalářská práce se zabývá segmentací obrazu s využitím metod založených na grafových řezech, konkrétně metodami Graph cut, Grab cut a One cut. V práci je uveden teoretický základ jak segmentace obrazu jako takové, tak těchto tří segmentačních metod. Kromě teoretického popisu byla v rámci této práce implementována aplikace, která provádí segmentaci pomocí těchto metod. Součástí práce je také otestování těchto segmentačních metod na sadě obrazů a porovnání jejich výsledků.

**Klíčová slova:** segmentace, One cut, Graph cut, Grab cut, OpenCV, Qt

## **Abstract**

This bachelor thesis deals with image segmentation using methods based on graph cuts, namely Graph cut, Grab cut and One cut. The thesis presents the theoretical basis for image segmentation and for the three segmentation methods. In addition to the theoretical description the application for image segmentation using the three methods is created. Part of the thesis is to test the three segmentation methods on a set of images and compare their results.

**Keywords:** segmentation, One cut, Graph cut, Grab cut, OpenCV, Qt

# Obsah

<b>Seznam obrázků</b>	<b>7</b>
<b>Seznam tabulek</b>	<b>8</b>
<b>1 Úvod</b>	<b>9</b>
<b>2 Segmentace obrazu</b>	<b>10</b>
2.1 Využití segmentace obrazu . . . . .	10
2.2 Segmentační metody . . . . .	12
<b>3 Segmentace obrazu s využitím grafových řezů</b>	<b>15</b>
3.1 Graph Cut . . . . .	15
3.2 Grab Cut . . . . .	17
3.3 One Cut . . . . .	19
<b>4 Aplikace pro segmentaci obrazu</b>	<b>22</b>
4.1 Obecný popis aplikace . . . . .	22
4.2 Implementace Graph cut . . . . .	23
4.3 Implementace metody One cut . . . . .	25
4.4 Implementace metody Grab cut . . . . .	26
4.5 Výpočet váhy hran v grafu . . . . .	27
<b>5 Porovnání metod Graph cut, One cut a Grab cut</b>	<b>28</b>
5.1 Nastavení parametrů metod při testování aplikace . . . . .	31
5.2 Porovnání metod inicializovaných pomocí ohraničení . . . . .	31
5.3 Porovnání metod inicializovaných pomocí vzorků . . . . .	32
5.4 Porovnání metod inicializovaných ohraničením a metod inicializovaných pomocí vzorků . . . . .	34
5.5 Testování segmentačních metod na obrazech bez odstranění šumu . . . . .	35
<b>6 Závěr</b>	<b>38</b>
<b>Literatura</b>	<b>39</b>

## Seznam obrázků

1	Volume rendering . . . . .	11
2	Postup detekce značek v obraze a umístění objektu do prostředí . . . . .	12
3	Ukázka prahování . . . . .	12
4	Ukázka Cannyho detektoru hran . . . . .	13
5	Ukázka segmentace Watershed . . . . .	14
6	Znázornění segmentace pomocí grafových řezů . . . . .	17
7	Ukázka grafu k funkci $E_{L1}$ . . . . .	20
8	Ukázka grafického rozhraní aplikace . . . . .	22
9	Ukázka výsledků segmentace . . . . .	29
10	Chybovost metody One cut . . . . .	30
11	Doba běhu metody One cut . . . . .	30
12	Ukázka segmentace One cut s malým počtem barevných skupin . . . . .	31
13	Porovnání Grab cut a One cut . . . . .	32
14	Porovnání Grab cut a One cut . . . . .	32
15	Porovnání Graph cut a One cut . . . . .	33
16	Ukázka výsledků segmentace metodou Grab cut pro dvě různá označení pomocí vzorků . . . . .	34
17	Ukázka výsledků segmentace s nevhodným označením objektu pomocí vzorků . .	35
18	Porovnání výsledků segmentace pomocí metody One cut . . . . .	35
19	Ukázka obrázku znečištěného šumem . . . . .	36

## Seznam tabulek

1	Porovnání výsledků segmentačních metod . . . . .	28
2	Porovnání segmentačních metod na obrazech, které nebyly rozmazány Gaussovým filtrem . . . . .	36
3	Porovnání segmentačních metod na obrazech, které nebyly rozmazány Gussovým filtrem a před zpracováním byly poškozeny šumem . . . . .	36



# 1 Úvod

Segmentace obrazu je jedním z prvních kroků v oblasti zpracování obrazu. Jedná se o proces, který rozděluje pixely obrazu do různých oblastí na základě určitého kritéria homogenity (např. jas). Vzniklé oblasti se nesmí překrývat a dohromady musí tvořit celý obraz. Doposud bylo vyvinuto velké množství segmentačních algoritmů s cílem dosáhnout ideální segmentace. V případě ideální segmentace jedna oblast představuje jeden objekt bez jakýchkoliv chyb. V praxi se však zcela ideální segmentace nedá dosáhnout, protože to vyžaduje porozumění obrazu jako celku a ne jen na úrovni pixelů, což zatím žádná segmentační metoda neumí. Ideální segmentaci se dá však hodně přiblížit, k tomu se využívají pro různé skupiny obrázků různé segmentační metody, nebo kombinace těchto metod. Cílem segmentace je změnit reprezentaci obrazu tak, aby bylo možné jej jednodušeji analyzovat. Výsledkem segmentace může být buďto nějaký hledaný objekt, nebo například jen množina přímek, popisujících hranici objektů.

Cílem této práce je představit a popsat základní segmentační metody a metody založené na grafových řezech, konkrétně metody Graph cut [2], Grab cut [3] a One cut [1]. Metoda Graph cut je první metoda využívající k segmentaci grafových řezů. Tato metoda vyžaduje od uživatele označení některých pixelů patřících objektu a některých pixelů patřících pozadí pomocí vzorků. Algoritmus pak převede obraz na graf, kde určí váhy jednotlivých hran a pomocí minimálního řezu grafu určí výslednou segmentaci obrazu. Metoda Grab cut vychází z metody Graph cut, ale je robustnější a není tolik citlivá na uživatelem zadávaná data. Metoda One cut je z těchto metod tou nejmladší. Tato metoda může segmentovat obraz jak pomocí objektu označeného ohrazením, tak pomocí objektu a pozadí označených pomocí vzorků. Dalším cílem této práce je implementace aplikace, která bude provádět segmentaci pomocí těchto tří segmentačních metod a porovnání výsledků segmentace na sadě reálných obrazů.

Druhá kapitola této práce obsahuje stručný úvod do segmentace obrazu. V této kapitole je popsáno několik odvětví, kde se využívá segmentace obrazu a je zde popsáno několik základních segmentačních metod.

Třetí kapitola je věnovaná segmentaci obrazu s využitím grafových řezů. V této kapitole jsou podrobně popsány výše zmíněné segmentační metody Graph cut, Grab cut a One cut.

Ve čtvrté kapitole je popsána aplikace pro segmentaci obrazu, jejíž vytvoření bylo také součástí této práce. Tato kapitola obsahuje obecný popis aplikace a popisy implementace jednotlivých segmentačních metod.

Pátá kapitola obsahuje porovnání výsledků segmentačních metod Graph cut, One cut a Grab cut na sadě reálných obrazů.

## 2 Segmentace obrazu

Segmentace obrazu je rozdělení obrazu na několik oblastí. Každá oblast pak obsahuje pixely s podobnými vlastnostmi, například jasnem, barvou nebo texturou. Pixely z různých oblastí mají tyto vlastnosti odlišné. Výsledkem segmentace mohou být buď jednotlivé oblasti, které musí pokrývat celou plochu obrázku, nebo jen obrys objektu. Segmentace hraje zásadní roli v oblasti počítačového vidění, rozpoznávání objektů a analýze obrazu. Segmentaci lze rozdělit na segmentaci automatickou, kam patří například segmentace Watershed [4], a segmentaci poloautomatickou, která vyžaduje označení částí jednotlivých segmentů uživatelem. Mezi poloautomatické segmentační metody patří například segmentační metody založené na grafových řezech.

### 2.1 Využití segmentace obrazu

Segmentace obrazu se využívá v mnoha oborech. S tím jak jsou vyvíjeny nové, dokonalejší metody segmentace, tak se segmentace obrazu využívá ve stále větším množství oborů.

#### 2.1.1 Biometrie

Jedním z nejrozšířenějších odvětví, kde se používá segmentace obrazu, je biometrie. Biometrie je metoda autentizace založená na rozpoznávání unikátních biologických charakteristik člověka [5]. Metody autentizace založené na segmentaci jsou:

- Detekce otisků prstu [6]
- Detekce duhovky [7]
- Detekce sítnice [8]

#### 2.1.2 Zpracování lékařských snímků

Segmentace obrazu je jednou z nejdůležitějších částí zpracování lékařských snímků. Jedná se o automatickou nebo poloautomatickou segmentaci, která na snímcích hledá nějaké neobvyklosti (např. nádor), nebo jen segmentuje hledané oblasti (např. kosti, nebo tkáň). Segmentují se různé typy snímků, ať už se jedná o snímky z počítačové tomografie, rentgenové snímky, nebo snímky z magnetické rezonance, kde je vzhledem k velkému množství snímků segmentace nutná [9]. Segmentace snímků z magnetické rezonance a počítačové tomografie se používá při tvorbě 3D modelů tkání. Tyto 3D modely pak umožňují zkoumání segmentovaných tkání z libovolného pohledu [4]. Na obrázku 1 je ukázka vizualizace CT snímku předloktí pomocí volume renderingu.



Obrázek 1: Vizualizace CT snímku předloktí pomocí volume renderingu [10]

### 2.1.3 Automobilový průmysl

V současnosti je jednou z nejdiskutovanějších problematik automobilového průmyslu autonomní řízení [11]. V tomto odvětví se pro segmentaci jednotlivých objektů z obrazu čím dál častěji používají konvoluční neuronové sítě [12] [13], které ve zpracování obrazu dosahují velmi dobrých výsledků.

Konvoluční neuronové sítě jsou podmnožinou neuronových sítí. Pro konvoluční sítě jsou typické konvoluční a pooling vrstvy, které umožňují konvolučním neuronovým sítím zpracovávat vstupní data na základě mnohem menšího množství parametrů. Díky těmto vrstvám, jsou tedy konvoluční neuronové sítě schopné zpracovávat i obrazy s vysokým rozlišením.

### 2.1.4 Rozšířená realita

Dalším odvětvím, ve kterém se používá segmentace obrazu, je rozšířená realita [4]. Rozšířená realita, na rozdíl od virtuální reality, nevytváří kolem uživatele umělé prostředí, ale rozšiřuje pohled na reálné prostředí. Do tohoto pohledu jsou pak vkládány jednotlivé objekty a je tak zachována realističnost prostředí. Pro určení kde v prostoru má být objekt umístěn, se dříve používaly speciální značky (markery), které pak byly v obraze detekovány. Postup detekce značek a umístění objektů do prostředí je znázorněn na obrázku 2. Pro detekci značek v obraze se pak využívala segmentace obrazu. Moderní knihovny už dokáží detekovat přímo nějaký 3D model a nepotřebují tedy žádné markery. Mezi tyto knihovny patří například i knihovna Vuforia [14]. Pomocí databáze vytvořené na základě 3D modelu objektu, který chceme v obraze detekovat, je pak knihovna schopná objekt v obraze najít, sledovat a dále s ním pracovat, např. přidávat popisky, nebo vkládat nové objekty.

Příkladem, kdy může být rozšířená realita použita, je asistenční software, např. pro pozemní personál letiště [15]. S použitím brýlí pro rozšířenou realitu bude zaměstnanec schopen skenovat identifikační kódy zavazadel a nákladu, které mu poskytnou informace o váze, pořadí nakládky a umístění nákladu v letadle.



Obrázek 2: Postup detekce značek v obraze a umístění objektu do prostředí [4]

## 2.2 Segmentační metody

Existuje mnoho segmentačních metod, žádná z nich však není univerzální. Většinou se nepoužívá jen jedna metoda, ale kombinace více metod. Metody segmentace obrazu můžeme rozdělit do tří základních skupin:

- Segmentační metody založené na analýze pixelů
- Segmentační metody založené na hledání hran
- Segmentační metody založené na detekci oblastí

### 2.2.1 Analýza pixelů

Jedná se o jednu z nejjednodušších a zároveň také velmi rychlou metodu segmentace. Typickým zástupcem této metody je prahování [16]. V této metodě jsou pixely rozdělovány na základě prahové hodnoty jejich jasu. Pokud je jas pixelu vyšší, než je prahová hodnota, je pixel detekován jako pixel patřící do hledané oblasti. Pokud je jeho jas nižší než prahová hodnota, je pixel detekován jako pixel patřící do pozadí. Výsledkem této segmentace je tedy binární obraz, kde černé pixely reprezentují objekt a bílé reprezentují pozadí. Efektivita prahování závisí na znalosti správné hodnoty jasu. Výhodou prahování je rychlost, nevýhodou je vysoká citlivost na šum. Existuje několik způsobů prahování např. hysterezní prahování [17]. Ukázka segmentace pomocí prahování je na obrázku 3.



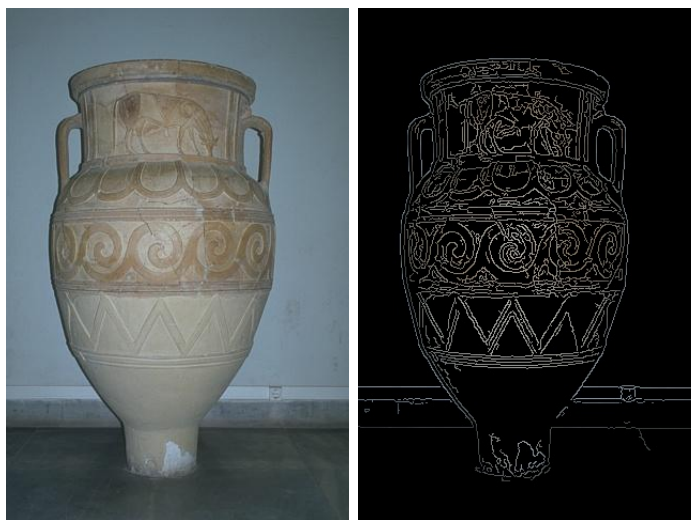
Obrázek 3: Ukázka prahování s automaticky nastaveným prahem [18]

### 2.2.2 Hledání hran

Segmentační metody založené na hledání hran představují velkou skupinu segmentačních algoritmů, přičemž každý z nich přistupuje k hledání hran jinak a každý z nich tak poskytuje jiný výsledek. Nejjednodušším způsobem, jak nalézt hrany v obraze, je zaměřit se na oblasti, kde se náhle mění jas. Na této myšlence jsou založeny gradientní metody hledání hran. Existuje velké množství algoritmů pro hledání hran, typickým zástupcem této metody je například Cannyho detektor hran [19].

#### Gradientní metody hledání hran

Tyto metody využívají k nalezení hrany první derivaci. Využívají toho, že v oblasti, kde se náhle mění jas, dosahuje absolutní hodnota první derivace vysokých hodnot. Výsledkem první derivace je pak hranový gradient, ten je následně porovnán s prahem, který určí, jestli se jedná o hranu, či nikoliv. K výpočtu gradientu lze také použít hranové operátory. V tomto případě se k výpočtu gradientu přistupuje jako ke konvolučnímu filtrování obrazu. Konvoluce se pak provádí pomocí konvolučního jádra zvoleného hranového operátoru. Těchto hranových operátorů bylo vyvinuto mnoho, mezi typické zástupce patří například Sobelův operátor, Kirschův operátor, Operátor Prewittové, nebo Robertsův operátor [4]. Výhodou gradientní metody je její rychlost, nevýhodou je vysoká citlivost na šum. Problém vysoké citlivosti na šum řeší Cannyho detektor hran. Výsledek detekce hran pomocí Cannyho detektoru je na obrázku 4.



Obrázek 4: Ukázka Cannyho detektoru hran [18]

### 2.2.3 Detekce oblastí

Segmentační metody založené na detekci oblastí rozdělují pixely do jednotlivých oblastí na základě určitého kritéria homogenity např. jasu, barvy, pokrytí stejnou texturou [20]. V porovnání

s metodou detekce hran se jedná o spolehlivější metodu v případě obrazů poškozených šumem. Tyto metody lze rozdělit do dvou základních skupin:

1. Detekce narůstáním oblastí
2. Detekce dělením oblastí

### **Detekce narůstáním oblastí**

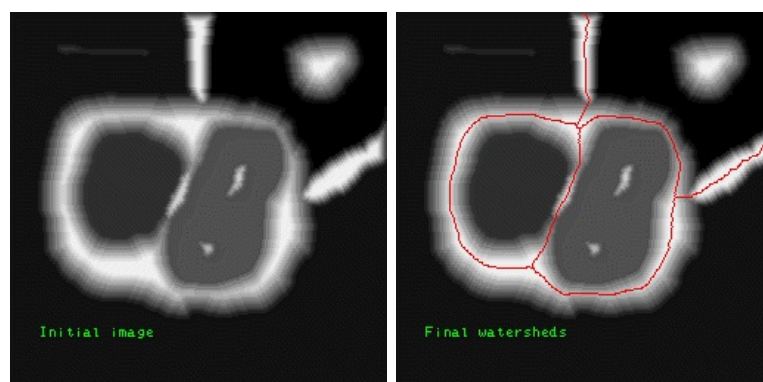
Tato metoda začíná pracovat s malými oblastmi, nebo i se samotnými pixely, a postupně je spojuje do menšího množství větších oblastí. Spojování jednotlivých oblastí probíhá do té doby, dokud je co spojovat. Dvě sousedící oblasti jsou spojeny do jedné, pokud mají stejnou barvu, jas, texturu a pokud mezi nimi není zřetelná hrana.

### **Detekce dělením oblastí**

Tato metoda postupuje obráceně než detekce narůstáním oblastí. Začíná tedy pracovat s jednou velkou oblastí, kterou postupně dělí, dokud nejsou splněna všechna kritéria homogenity.

### **Segmentace Watershed**

Segmentace Watershed [4] je metoda, patřící do skupiny metod založených na detekci oblastí. Tato segmentační metoda pracuje s obrazem jako s topologickým povrchem, kde je jas každého pixelu vnímán jako výška v daném bodě. Segmentace probíhá tak, že algoritmus začne všechna lokální minima zaplavovat vodou. V momentě kdy hrozí, že voda začne spojovat dvě sousední lokální minima, je mezi nimi vybudována hráz. Oblast lokálního minima je chápána jako segment, jednotlivé segmenty jsou pak odděleny hrázemi. Jedná se o osvědčenou a efektivní metodu segmentace. Nevýhodou této metody je vysoká citlivost na šum, kdy při vysokém šumu produkuje tato metoda velké množství vzniklých segmentů, které je nutné dále zpracovat. Segmentace rozvodím se hojně používá například v lékařství, k segmentaci snímků mozku. Ukázka segmentace Watershed je na obrázku 5.



Obrázek 5: Ukázka segmentace Watershed [4]

### 3 Segmentace obrazu s využitím grafových řezů

V předchozí kapitole jsem jednu ze segmentačních metod záměrně vynechal. Jedná se metodu segmentace obrazu s využitím grafových řezů, a této metodě se budu podrobněji věnovat v této kapitole. Segmentaci pomocí grafových řezů poprvé představili Boykov a Jolly v roce 2001 v článku [2]. Od té doby bylo vyvinuto mnoho dalších segmentačních algoritmů založených na grafových řezech, já se ve své práci budu věnovat následujícím třem: Graph Cut [2], One Cut [1], Grab Cut [3].

#### 3.1 Graph Cut

Jedná se o první segmentační metodu založenou na grafových řezech. Tento algoritmus poprvé představili Yuri Boykov a Marie-Pierre Jolly v roce 2001 [2]. Jedná se interaktivní segmentační metodu, to znamená, že uživatel musí označit některé pixely patřící objektu a některé pixely patřící pozadí. Takto označené pixely se nazývají jako tvrdá omezení. Z těchto tvrdých omezení jsou pak vypočítány histogramy pro objekt a pozadí. Další, tzv. měkká omezení, vycházejí z vlastností hran a oblastí.

Nechť  $P$  je množina všech pixelů v obraze a  $N$  je množina párů všech sousedících pixelů z  $P$ . Dále uvažujme vektor  $A = (A_1, \dots, A_p, \dots, A_{|P|})$ , kde jednotlivé komponenty  $A_p$  určují zařazení pixelů  $p$  z  $P$  do objektu, nebo do pozadí.

Měkká omezení, která vycházejí z vlastností hran a oblastí, jsou popsána cenovou funkcí  $E(A)$ :

$$E(A) = \lambda \cdot R(A) + B(A) \quad (1)$$

kde

$$R(A) = \sum_{p \in P} R_p(A_p) \quad (2)$$

$$B(A) = \sum_{\{p,q\} \in N} B_{\{p,q\}} \cdot \delta(A_p, A_q) \quad (3)$$

a

$$\delta(A_p, A_q) = \begin{cases} 1, & \text{pokud } A_p \neq A_q \\ 0, & \text{jinak.} \end{cases} \quad (4)$$

Koeficient  $\lambda \geq 0$  v rovnici 1 určuje důležitost vlastností oblasti  $R(A)$  oproti vlastnostem hranice  $B(A)$ . Člen  $R(A)$  předpokládá, že už jsou dány jednotlivé penalizace za zařazení pixelu  $p$  do objektu  $R_p = ("obj")$  nebo pozadí  $R_p = ("bkg")$ .  $R_p = (\cdot)$  může například reagovat na to, jak pixel  $p$  zapadá do histogramu pozadí a objektu.

Koeficient  $B(A)$  zahrnuje vlastnosti hranice segmentace  $A$ . Koeficient  $B_{\{p,q\}} \geq 0$  popisuje nesouvislost mezi pixely  $p$  a  $q$ . Standardně dosahuje koeficient  $B_{\{p,q\}} \geq 0$  vysoké hodnoty, pokud jsou pixely  $p$  a  $q$  podobné (například mají podobný jas). Naopak, pokud se pixely  $p$  a  $q$  hodně liší, blíží se tento koeficient nule. Pro výpočet cen  $B_{\{p,q\}}$  se může použít například některý z gradientních operátorů.

### 3.1.1 Segmentace

Mějme dvě množiny  $O$  a  $B$ , kde množina  $O$  představuje podmnožinu pixelů označených jako pixely objektu a množina  $B$  představuje podmnožinu pixelů označených jako pixely pozadí. Pro segmentaci pak platí:

$$\forall p \in O, A_p = \text{"obj"} \quad (5)$$

$$\forall p \in B, A_p = \text{"bkg"} \quad (6)$$

Jelikož se jedná o segmentační metodu, využívající k segmentaci řez grafu, je nutné obraz převést na graf. Vytvoříme tedy graf  $G = \langle V, E \rangle$ , kde  $V$  je množina všech vrcholů grafu a  $E$  je množina všech hran v grafu. Vrcholy grafu se pak dále dělí na vrcholy, které reprezentují jednotlivé pixely obrazu a na vrcholy, které se nazývají terminály. Terminály se skládají ze dvou vrcholů: vrchol  $S$  (pro objekt) a vrchol  $T$  (pro pozadí). Pro množinu hran tedy platí  $V = P \cup \{S, T\}$ . Takovýto graf se označuje jako s-t graf. Hrany grafu se rovněž dělí na dva druhy. Prvním druhem jsou hrany spojující sousedící vrcholy grafu, reprezentující pixely (n-hrany). Druhým druhem jsou hrany spojující jednotlivé vrcholy reprezentující pixely s terminály (t-hrany). Jednotlivým hranám  $e \in E$  jsou poté přiděleny nezáporné ceny  $\omega_e$ . Každý pixel  $p$  je spojen s oběma terminály pomocí dvou t-hran  $\{p, S\}$   $\{p, T\}$  a každý pár sousedících pixelů  $\{p, q\}$  je spojen pomocí n-hrany. Náhled takového grafu je na obrázku 6. Pro množinu hran tedy platí:

$$E = N \bigcup_{p \in P} \{\{p, S\}, \{p, T\}\} \quad (7)$$

Váha jednotlivých hran je dána následující tabulkou:

hrana	cena	pro
$\{p, q\}$	$B_{\{p,q\}}$	$\{p, q\} \in N$
$\{p, S\}$	$\lambda \cdot R_p(\text{"bkg"})$ $K$ $0$	$p \in P, p \notin O \cup B$ $p \in O$ $p \in B$
$\{p, T\}$	$\lambda \cdot R_p(\text{"obj"})$ $0$ $K$	$p \in P, p \notin O \cup B$ $p \in O$ $p \in B$

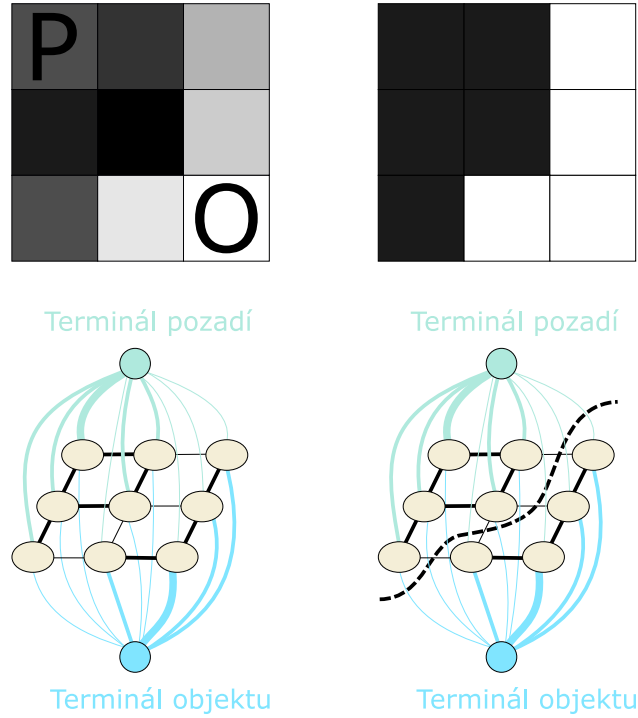


kde

$$K = 1 + \max_{p \in P} \sum_{q: \{p,q\} \in N} B_{\{p,q\}} \quad (8)$$

Nyní už máme graf  $G$  kompletně definovaný a můžeme provést segmentaci pomocí minimálního řezu grafu. Celý postup segmentace je znázorněn na obrázku 6. Pro provedení minimálního řezu existuje mnoho algoritmů, z nichž asi nejvyužívanější je algoritmus Boykov Kolmogorov [21]. Výsledkem řezu je podmnožina hran  $C \cup E$  taková, že terminál objektu a terminál pozadí jsou od sebe odděleny. Řez grafu minimalizuje energetickou funkci 1. Cena řezu je pak rovna součtu cen všech přerušených hran.

$$|C| = \sum_{e \in C} w_e \quad (9)$$



Obrázek 6: Znázornění segmentace pomocí grafových řezů

### 3.2 Grab Cut

Další segmentační metodou, která je založená na grafových řezech, je Grab Cut [3]. Tato metoda vychází z Graph Cut, ale je robustnější a není tolik citlivá na uživatelem zadaná data. Uživatel na vstupu označí oblast, ve které se nachází hledaný objekt obdélníkem a poskytne tak trimap  $T$ . Trimap  $T$  se skládá ze tří oblastí:  $T_B$  pro pozadí,  $T_F$  pro objekt  $T_U$  pro nezařazené pixely. Obraz je pak reprezentován jako pole jasů jednotlivých pixelů  $z = (z_1, \dots, z_n, \dots, z_N)$ . Segmentaci

představuje pole hodnot  $\underline{\alpha} = (\alpha_1, \dots, \alpha_N)$ , kde se každé  $\alpha_n$  vztahuje k jednomu pixelu  $z_n$  se stejným indexem  $n$ . Obvykle platí  $0 \leq \alpha_n \leq 1$ , ale pro tvrdou segmentaci se používá  $\alpha_n \in \{0, 1\}$ , kde 0 indikuje, že pixel patří do pozadí a 1 indikuje, že pixel patří do objektu. Parametr  $\theta$  popisuje rozdělení barev pro pozadí a objekt, a skládá se ze dvou modelů směsi gaussovských funkcí (GMM), jeden pro pozadí a jeden pro objekt. Každý model se skládá z  $K$  komponent (obvykle  $K = 5$ ). Vzhledem k poddajnosti GMM je zaveden dodatečný vektor  $\mathbf{k} = \{k_1, \dots, k_n, \dots, k_N\}$ , kde  $k_n \in \{1, \dots, K\}$ . Tento vektor přiřazuje každému pixelu unikátní GMM komponentu, buď z pozadí nebo z objektu, v závislosti na tom, jestli  $\alpha_n = 0$  nebo 1. Energie pro segmentaci je pak ve tvaru:

$$\mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}, z) = U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, z) + V(\underline{\alpha}, z) \quad (10)$$

Datový člen  $U$  je definován jako:

$$U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, z) = \sum_n D(\alpha_n, k_n, \underline{\theta}, z_n) \quad (11)$$

kde  $D(\alpha_n, k_n, \underline{\theta}, z_n) = -\log p(z_n | \alpha_n, k_n, \underline{\theta}) - \log \pi(\alpha_n, k_n)$ , kde  $p(\cdot)$  je Gaussovo rozdělení pravděpodobnosti a  $\pi(\cdot)$  jsou váhy koeficientů GMM. Datový člen je tedy možné zapsat jako:

$$\begin{aligned} D(\alpha_n, k_n, \underline{\theta}, z_n) = & -\log \pi(\alpha_n, k_n) + \frac{1}{2} \log \det \Sigma(\alpha_n, k_n) + \\ & + \frac{1}{2} [z_n - \mu(\alpha_n, k_n)]^T \Sigma(\alpha_n, k_n)^{-1} [z_n - \mu(\alpha_n, k_n)] \end{aligned} \quad (12)$$

Parametry modelu pak vypadají následovně:

$$\underline{\theta} = \{\pi(\alpha, k), \mu(\alpha, k), \Sigma(\alpha, k), \alpha = 0, 1, k = 1 \dots K\} \quad (13)$$

Hladkostní člen  $V$  může být zapsán ve tvaru:

$$V(\underline{\alpha}, \mathbf{z}) = \gamma \sum_{(m,n) \in \mathbf{C}} [\alpha_n \neq \alpha_m] \exp -\beta \|z_m - z_n\|^2 \quad (14)$$

kde  $\mathbf{C}$  je množina sousedících pixelů. Pokud je konstanta  $\beta = 0$ , zajišťuje hladkostní člen hladkost všude, do úrovně omezené konstantou  $\gamma$ . Je však mnohem efektivnější nastavit  $\beta > 0$ , což sníží tendenci spojovat oblasti s velkými kontrastními rozdíly. Konstanta  $\beta$  se vybírá tak, aby splňovala:

$$\beta = \left( 2 \left\langle (z_m - z_n)^2 \right\rangle \right) \quad (15)$$

Tato volba  $\beta$  zajišťuje, aby hladkostní člen správně přecházel mezi oblastmi s vysokým a nízkým kontrastem.

### 3.2.1 Segmentace

Segmentace je určena jako globální minimum:

$$\hat{\underline{\alpha}} = \arg \min_{\underline{\alpha}} \mathbf{E}(\underline{\alpha}, \underline{\theta}) \quad (16)$$

Minimalizace energie probíhá iterativně, což umožňuje automatické změny hodnot  $\underline{\alpha}$  v závislosti na tom, jak jsou nově nalezené pixely z oblasti  $T_U$  použity ke změně parametrů GMM. Segmentace se skládá z následujících kroků:

1. Uživatel poskytne pozadí  $T_B$  a tím inicializuje T. Popředí je pak nastaveno na  $T_F = \emptyset$  a  $T_U$  je nastaveno jako doplněk pozadí  $T_U = \overline{T}_B$ .
2. Nastavení  $\alpha_n = 0$  pro  $n \in T_B$  a  $\alpha_n = 1$  pro  $n \in T_U$
3. Inicializace obou modelů GMM, na základě hodnot  $\alpha$
4. Přiřazení komponentů modelu GMM pixelům. Pro každé  $n$  v oblasti  $T_U$

$$k_n := \arg \min_{k_n} D_n(\alpha_n, k_n, \theta, z_n)$$

5. Výpočet nových parametrů GMM pomocí dat  $z$ :

$$\underline{\theta} := \arg \min_{\underline{\theta}} (\underline{\alpha}, \mathbf{k}, \underline{\theta}, z)$$

6. Určení řezu pomocí minimálního řezu:

$$\min_{\alpha_n: n \in T_U} \min_k \mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}, z)$$

7. Opakování od kroku 4, dokud není dosažena konvergence.

## 3.3 One Cut

One Cut [1] je segmentační metoda, která vychází z Grab Cut. Odstraňuje nutnost iterativní minimalizace energie a tudíž je rychlejší a poskytuje i lepší výsledky. Tato metoda rozděluje interval hodnot pixelů  $H = \langle H_{\min}, H_{\max} \rangle$  do  $K$  menších intervalů (barevných skupin)  $B_1, B_2, \dots, B_K$ . Metoda One Cut také zavádí člen  $L_1$ , který měří překrytí mezi segmenty.

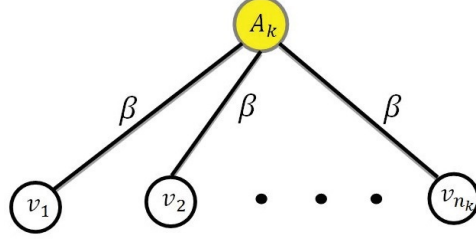
### 3.3.1 Minimalizace překrytí vzhledů

Nechť  $P$  je množina všech pixelů obrazu a  $S \subset \Omega$  je segment a necht  $\theta^S$  a  $\theta^{\overline{S}}$  jsou nenormalizované histogramy pozadí a objektu. Dále mějme  $n_k$  určující počet pixelů v intervalu  $B_k$  a  $n_k^S$  a  $n_k^{\overline{S}}$

určující počet pixelů v objektu a pozadí pro interval  $B_k$ . Průnik intervalů pozadí a objektu je penalizován zahrnutím členu  $L_1$  do funkce energie:

$$E_{L_1}(\theta^S, \theta^{\bar{S}}) = -\|\theta^S - \theta^{\bar{S}}\|_{L_1} = \sum_{k=1}^K \min(n_k^S, n_k^{\bar{S}}) - \frac{|\Omega|}{2} \quad (17)$$

Ukázka grafu pro funkci  $E_{L_1}$  je na obrázku 7.



Obrázek 7: Ukázka grafu k funkci  $E_{L_1}$ . Uzly  $v_1, \dots, v_{n_k}$  reprezentují jednotlivé pixely obrazu, které patří do jedné barevné skupiny  $B_k$ . Tyto uzly jsou propojeny s uzlem  $A_k$ , který představuje danou barevnou skupinu  $B_k$ , pomocí neorientovaných hran, jejichž kapacita odpovídá váze členu překrytí  $\beta > 0$  [1].

Do grafu přidáme  $k$  pomocných uzlů  $A_1, A_2, \dots, A_K$ , přičemž každý uzel bude reprezentovat jeden interval histogramu. Každý uzel  $A_k$  propojíme se všemi uzly, které reprezentují pixely patřící do intervalu s indexem  $k$ . Kapacita těchto hran je nastavena na  $\beta = 1$ . Nyní uvažujme, že nějaký interval s indexem  $k$  obsahuje jak pixely patřící do pozadí, tak pixely patřící do objektu. V takovém případě musí být při oddělení pozadí od objektu přerušeno  $n_k^S$  nebo  $n_k^{\bar{S}}$  hran, spojujících dané uzly s pomocným vrcholem  $A_k$ . Optimální řez tedy musí vybrat  $\min(n_k^S, n_k^{\bar{S}})$ .

### 3.3.2 Segmentace pomocí ohraničení

Uživatel poskytne obdélník definující oblast, ve které se nachází hledaný objekt, stejně jako v metodě Grab Cut. Pixely mimo tento obdélník jsou, pomocí tvrdých omezení, označeny jako pixely patřící do pozadí. Necht  $\Omega$  je množina všech pixelů obrazu,  $R \subseteq \Omega$  je binární maska reprezentující uživatelem zadanou oblast,  $S_{GT} \subseteq \Omega$  je ideální segmentace a  $S \subseteq \Omega$  je segment. Energie pro segmentaci je pak ve tvaru:

$$E(S) = |\bar{S} \cap R| - \beta \|\theta^S - \theta^{\bar{S}}\|_{L_1} + \lambda |\partial S| \quad (18)$$

kde  $\lambda |\partial S|$  je hladkostní člen citlivý na kontrast. Pro výpočet hladkostního členu se používá  $|\partial S| = \sum \omega_{pq} |s_p - s_q|$  kde  $\omega_{pq} = \frac{1}{\|p-q\|} \cdot e^{\frac{-\delta I^2}{2\sigma^2}}$  kde  $\sigma^2$  je průměrné  $\Delta I^2$  v celém obraze. Optimalizace této energie je provedena jedním grafovým řezem. Váha energie se běžně upravuje podle vstupního obrazu a daného ohraničení. Aby se váha  $\beta$  upravovala automaticky, je možné použít člen překrytí

vzhledu. Specifické  $\beta_{Img}$  pro daný obraz je pak možné zapsat ve tvaru:

$$\beta_{Img} = \frac{|R|}{-\|\theta_R - \theta_{\bar{R}}\|_{L_1} + |\Omega|/2} \cdot \beta' \quad (19)$$

kde  $\beta'$  je globální parametr.

### 3.3.3 Segmentace s využitím vzorků

Tato varianta metody One Cut má na vstupu obraz, kde jsou označeny pixely objektu a pozadí, stejně jako v metodě Graph Cut. Díky tomu, že tato metoda nevyužívá k označení objektu obdélník, jako předchozí metoda, je energie pro segmentaci jednodušší:

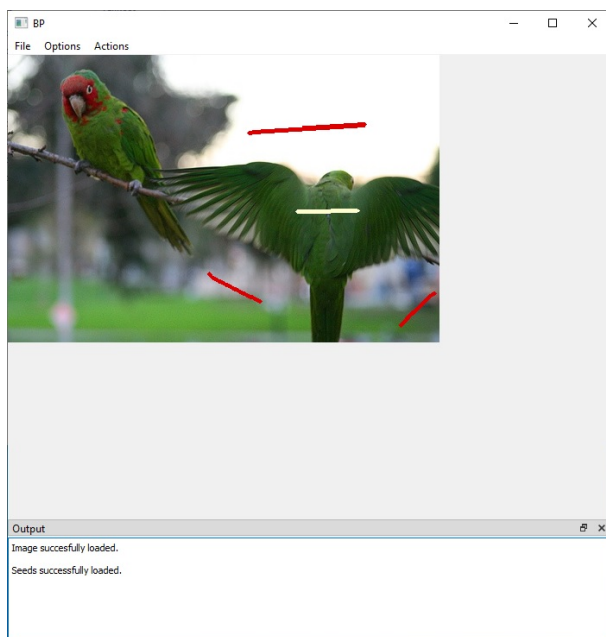
$$E_{seeds}(S) = -\beta\|\theta^S - \theta^{\bar{S}}\|_{L_1} + \lambda|\partial S| \quad (20)$$

## 4 Aplikace pro segmentaci obrazu

Další částí mé bakalářské práce byla tvorba aplikace pro segmentaci obrazu pomocí metod Graph Cut, One Cut a Grab Cut. Tato kapitola bude věnována popisu této aplikace a popisu implementace jednotlivých segmentačních metod.

### 4.1 Obecný popis aplikace

Aplikace pro segmentaci obrazu je vytvořena v jazyce C++ s použitím knihovny OpenCV, která umožňuje práci s obrazy a knihovny Qt, s jejíž pomocí je vytvořeno grafické rozhraní aplikace. Ukázka grafického rozhraní aplikace je na obrázku 8. Grafické rozhraní aplikace umožňuje uživateli načíst obraz, který bude segmentovat, obraz s ideální segmentací, obraz se vzorky a obraz s ohrazením. Aplikace dává uživateli rovněž možnost zakreslit vzorky i ohrazení přímo do načteného obrazu pro segmentaci. Pokud je načtený obraz pro segmentaci a jsou taktéž načteny, nebo vytvořeny vzorky, nebo ohrazení, může uživatel spustit segmentaci pomocí jedné z výše zmíněných metod.



Obrázek 8: Ukázka grafického rozhraní aplikace

Aplikace také umožňuje provést segmentaci celé sady obrazů. V tomto případě musí být každý segmentovaný obraz ve vlastní složce a musí být pojmenovaný `img.jpg`. V této složce musí být také obraz se vzorky, pojmenovaný `seeds.png`, dále obraz s ohrazením, pojmenovaný `box.bmp` a obraz s ideální segmentací, pojmenovaný `gt.png`. Po provedení segmentace celé sady obrazů vypíše aplikace chybovost jednotlivých segmentačních metod, společně se směrodatnou odchylkou, a taktéž časy těchto segmentačních metod včetně jejich směrodatné odchylky.

Chybovost pro metody inicializované pomocí vzorků je vypočítána jako:

$$ER = 100 \cdot \frac{P_e}{P - P_s} \quad (21)$$

kde  $P$  je počet pixelů obrazu,  $P_e$  je počet pixelů chybně zařazených buďto do objektu nebo do pozadí a  $P_s$  je počet pixelů označených pomocí vzorků. Chybovost pro metody inicializované ohraňováním je vypočítána jako:

$$ER = 100 \cdot \frac{P_e}{P_b} \quad (22)$$

kde  $P_b$  je počet pixelů v ohraňování.

Během segmentace jsou pro každý obraz průběžně ukládány výsledky jednotlivých segmentačních metod, vždy do složky, kde se obraz nachází. Aplikace se skládá z následujících částí:

- main - Je metoda, která spouští uživatelské rozhraní.
- GraphCut - Je třída, která provádí segmentaci obrazu pomocí metody Graph cut.
- OneCut - Je třída, která provádí segmentaci obrazu pomocí metody One cut. Umožňuje provádět segmentaci, kde je objekt označen vzorky i segmentaci, kde je objekt označen ohraňováním.
- GrabCut - Je třída, která umožňuje provádět segmentaci pomocí metody Grab cut. Tato třída používá pro segmentaci metodu Grab cut, implementovanou v knihovně OpenCV.
- ImageArea - Je třída, která dědí ze třídy QWidget, umožňuje načítání a ukládání obrazů a umožňuje uživateli zakreslovat do obrazu vzorky a ohraňování.
- MainWindow - Je třída, která dědí ze třídy QMainWindow. Tato třída používá instanci třídy ImageArea jako centrální widget, pomocí menubaru umožňuje ovládat aplikaci a používá instanci třídy QTextEdit pro zobrazování informací uživateli.
- DataSetSegmentation - Je třída, která zajišťuje segmentaci celé sady obrazů pomocí všech segmentačních metod, které aplikace nabízí.

Dále pak aplikace obsahuje třídy, které jsou součástí knihovny pro vytvoření grafu, práci s grafem a pro výpočet minimálního řezu. Tuto knihovnu poskytují autoři k článku [21].

## 4.2 Implementace Graph cut

V této kapitole se budu věnovat popisu implementace metody Graph Cut. Segmentace metodou Graph cut se dá popsat následujícím obecným algoritmem:

---

**Algoritmus 1** Segmentace metodou Graph Cut

---

- 1: Rozmaž obraz Gaussovým filtrem
  - 2: Urči váhy hran v grafu.
  - 3: Převed vstupní obraz na graf.
  - 4: Proveď minimální řez grafu.
  - 5: Pomocí výsledku minimálního řezu grafu odděl ve vstupním obraze pozadí od objektu.
- 

Pro rozmazání obrazu Gaussovým filtrem jsem využil metodu GaussianBlur, která je implementována v knihovně OpenCV. Rozmazání obrazu proběhne vždy a to za požití masky o velikosti 5x5, přičemž hodnoty  $\sigma_x$  i  $\sigma_y$  jsou nastaveny na hodnotu 0,7.

#### 4.2.1 Výpočet váhy hran

Jak už bylo zmíněno v kapitole věnované popisu metody Graph cut, obsahuje graf dva druhy hran. To jsou hrany spojující uzly grafu reprezentující sousedící pixely v obraze (n-hrany) a hrany, spojující jednotlivé uzly s terminály (t-hrany). Pro tyto dva typy hran je potřeba vypočítat jejich váhy. Váha n-hrany se vypočítá jako:

$$\omega_i = \lambda e^{\frac{-\|C_i - C_j\|^2}{2*\sigma}} \quad (23)$$

kde  $\lambda$  je konstanta,  $\sigma$  je vypočítána jako průměrná druhá mocnina Euklidovy vzdálenosti barev sousedících pixelů a  $\|C_i - C_j\|$  se vypočítá jako Euklidova vzdálenost barev daných pixelů.

T-hrany se dále rozdělují na dva typy. Jedním typem jsou hrany, spojující s terminály pixely, které jsou pomocí vzorků již označeny jako pixely patřící pozadí, nebo jako pixely patřící objektu. Druhým typem jsou hrany spojující s terminály pixely, které nejsou nijak označeny. U prvního typu těchto hran je určování jejich váhy jednoduché, a určuje se pomocí tzv. tvrdých omezení. Pokud je pixel označen jako pixel patřící objektu, je váha hrany spojující tento pixel s objektem  $\omega_{iF} = \infty$  (respektive vysoká konstanta) a váha hrany spojující tento pixel s pozadím je  $\omega_{iB} = 0$ . U pixelů označených jako pozadí, jsou tyto váhy prohozené, tedy  $\omega_{iF} = 0$  a  $\omega_{iB} = \infty$ . Váha pixelů, které nejsou nijak označeny, se určí jako:

$$\begin{aligned} \omega_{iF} &= -\log P_B(i) \\ \omega_{iB} &= -\log P_F(i) \end{aligned} \quad (24)$$

kde  $P_B(i)$  je pravděpodobnost zařazení pixelu  $i$  do pozadí,  $P_F(i)$  je pravděpodobnost zařazení pixelu  $i$  do objektu.

Pro výpočet pravděpodobnosti jsem použil model „Expectation Maximization“, který je naimplementován v knihovně OpenCV. Tyto modely jsem vytvořil dva, jeden pro objekt a druhý pro pozadí, a v každém modelu jsem pak nechal odhadnout gaussovskou směsici jasů pro objekt nebo pozadí z označených pixelů. Z těchto směsí jsem pak nechal Expectation Maximization



modely určit logaritmus pravděpodobnosti zařazení neoznačených pixelů do objektu a pozadí. Tuto hodnotu jsem pak použil pro určení váhy odpovídající hrany.

#### 4.2.2 Převod vstupního obrazu na graf a řez grafu

Jak už jsem zmínil výše, pro práci s grafem jsem použil knihovnu, kterou poskytují autoři k článku [21]. Při tvorbě grafu je nutné aby existovala vazba mezi uzlem grafu a odpovídajícím pixelem. ID každého uzlu grafu je tedy určeno následovně:

$$id = x_i + y_i \cdot w \quad (25)$$

kde  $x_i$  je pozice pixelu  $i$  na ose  $x$ , kde  $y_i$  je pozice pixelu  $i$  na ose  $y$  a  $w$  je šířka obrazu. Nastavil jsem odpovídající váhy hran a provedl minimální řez grafu. Podle výsledného grafu je pak barva pixelů, jejichž odpovídající uzly v grafu zůstaly propojeny pouze s terminálem pozadí, nastavena na bílou, aby se ve výsledném obrazu opticky oddělilo pozadí od objektu.

### 4.3 Implementace metody One cut

Další metodou segmentace, kterou aplikace provádí, je metoda One cut. Obecný algoritmus pro segmentaci metodou One cut je stejný jako u metody Graph cut, avšak váhy hran grafu a vytvořený graf se liší. Jelikož aplikace podporuje One cut segmentaci inicializovanou jak pomocí vzorků, tak pomocí ohrazení, liší se výpočet váhy hran i podoba výsledného grafu pro tyto dva typy One cut segmentace. Implementaci těchto dvou druhů inicializace rozeberu v následujících dvou podkapitolách.

#### 4.3.1 Vytváření grafu metody One cut inicializované pomocí vzorků

Prvním způsobem inicializace metody One cut je inicializace pomocí vzorků. V tomto případě jsou některé pixely objektu a některé pixely pozadí označeny pomocí vzorků stejně jako v metodě Graph cut. Jelikož metoda One cut využívá pomocné uzly, nejsou s terminály propojeny všechny uzly grafu. S terminály jsou propojeny jen uzly, reprezentující pixely označené pomocí vzorků jako pixely patřící do pozadí nebo jako pixely patřící objektu. Váha těchto hran bude určena tvrdými omezeními, stejně jako v metodě Graph cut. Váha hran, spojujících uzly reprezentující sousedící pixely, bude také stejná jako v metodě Graph cut, a váha hran spojujících uzly, představující pixely, s pomocnými uzly je konstantní.

#### 4.3.2 Vytváření grafu metody One cut inicializované ohrazením

Pokud je metoda One cut inicializovaná ohrazením, jsou s terminály propojeny všechny uzly grafu. Uzly reprezentující pixely, které se nacházejí vně ohrazení, jsou pomocí tvrdých omezení označeny jako pixely patřící do pozadí. Pixely nacházející se uvnitř ohrazení jsou spojeny s terminálem objektu hranou s konstantní váhou  $\omega_{iF} = 1$  a s terminálem pozadí hranou s

konstantní váhou  $\omega_{iB} = 0$ . Všechny uzly reprezentující sousedící pixely jsou spolu propojeny hranami se stejnou váhou jako v případě metody Graph cut a všechny uzly reprezentující pixely jsou opět propojeny s odpovídajícími pomocnými uzly. Váha těchto hran je pak určena pomocí členu  $L_1$ . Řez grafu a následná segmentace obrazu probíhá stejně jako v případě metody Graph cut.

## 4.4 Implementace metody Grab cut

Jelikož knihovna OpenCV má již metodu Grab cut naimplementovanou, využil jsem při tvorbě aplikace této implementace. Tato implementace Grab cut umožňuje inicializovat metodu jak pomocí ohraničení, tak pomocí vzorků.

### 4.4.1 Segmentace inicializovaná ohraničením

Metoda Grab cut implementovaná v knihovně OpenCV má na vstupu sedm parametrů. Prvním parametrem je matice RGB hodnot reprezentující vstupní obraz. Druhým parametrem je matice, která slouží pro předání vzorků v případě použití metody Grab cut inicializované pomocí vzorků. V tomto případě, kdy je metoda Grab cut inicializována pomocí ohraničení, se jako druhý parametr uvádí prázdná matice. Tato matice pak po provedení segmentace obsahuje informace o tom, který pixel byl při segmentaci označen jako pixel patřící do pozadí a který byl označen jako patřící do objektu. Třetím argumentem metody je obdélník představující ohraničení. Všechny pixely vně ohraničení jsou v metodě označeny jako pixely pravděpodobně patřící do pozadí a všechny pixely uvnitř ohraničení jsou označeny jako pixely pravděpodobně patřící do objektu. Dalšími dvěma parametry jsou matice představující model objektu a model pozadí. Tyto matice jsou interně využívány algoritmem a při předávání metodě se předávají prázdné matice. Šestý argument určuje počet iterací metody a poslední argument určuje způsob inicializace metody, tedy jestli je metoda inicializovaná pomocí vzorků, nebo pomocí ohraničení.

### 4.4.2 Segmentace inicializovaná pomocí vzorků

Jak už jsem zmínil výše, metoda Grab cut kterou implementuje knihovna OpenCV, může být inicializována jak pomocí vzorků, tak pomocí ohraničení. Inicializace pomocí vzorků probíhá obdobně, jako inicializace pomocí ohraničení, některé předávané parametry se však liší. Na pozici druhého parametru je metodě předána matice, reprezentující označení jednotlivých pixelů pomocí vzorků. Každý prvek této matice představuje jeden pixel obrazu a může nabývat čtyř hodnot. Tyto čtyři hodnoty jsou definovány jako konstanty v knihovně OpenCV a slouží k označení daného pixelu, jako pixelu patřícího pravděpodobně do pozadí, pravděpodobně do objektu, určitě do pozadí, nebo určitě do objektu. Pixely, které jsou označeny pomocí vzorků jsou pak v této matici označeny jako pixely patřící určitě do objektu, nebo jako pixely patřící určitě do pozadí. Všechny ostatní pixely v matici jsou označeny jako pixely patřící pravděpodobně

do pozadí. Jako třetí parametr je metodě předán objekt čtverce, kde jsou souřadnice i rozměry nastaveny na 0. V posledním parametru je pak potřeba zvolit inicializaci pomocí vzorků.

#### 4.5 Výpočet váhy hran v grafu

Váha  $n$ -hran se v metodě Grab cut vypočítá jako:

$$\omega_i = \gamma e^{-\beta \|C_i - C_j\|^2} \quad (26)$$

kde  $\gamma$  je konstanta,  $\|C_i - C_j\|$  se vypočítá jako Euklidova vzdálenost barev daných pixelů a  $\beta$  je konstanta, která se pro každý obraz vypočítá jako převrácená hodnota dvojnásobku průměrného rozdílu barev sousedících pixelů. Váhy  $t$ -hran se pak liší pro pixely, které jsou označeny jako pravděpodobně patřící do objektu nebo do pozadí a pro pixely označené jako určité patřící do objektu nebo určité patřící do pozadí. V prvním případě, kdy jsou pixely označeny jako pixely určité patřící do pozadí, je váha hran spojující tyto pixely s terminály určena pomocí tvrdých omezení. V druhém případě jsou váhy hran určeny jako:

$$\begin{aligned} \omega_{iF} &= -\log P_B(i) \\ \omega_{iB} &= -\log P_F(i) \end{aligned} \quad (27)$$

kde  $P_B(i)$  je pravděpodobnost zařazení pixelu  $i$  do pozadí,  $P_F(i)$  je pravděpodobnost zařazení pixelu  $i$  do objektu. Pro výpočet pravděpodobnosti jsou použity dva modely gaussovských směsí, jeden pro objekt a druhý pro pozadí. Gaussovská směs pro objekt je vytvořena z pixelů, označených jako pravděpodobně, nebo určité patřící do objektu, gaussovská směs pro pozadí je pak vytvořena z pixelů, označených jako pravděpodobně, nebo určité patřící do pozadí.

## 5 Porovnání metod Graph cut, One cut a Grab cut

Hlavní částí mé bakalářské práce bylo otestování segmentačních metod Graph cut, One cut a Grab cut na sadě reálných obrazů a zhodnocení dosažených výsledků. Pro otestování jsem použil sadu obrazů, kterou poskytují autoři k článku [22]. Tato sada obsahuje 151 obrazů o velikostech v rozmezí od 284 x 398 až 800 x 618 pixelů. Autoři článku k této sadě obrazů také poskytují sadu ideálních segmentací a sadu souborů pro inicializaci pomocí vzorků. Pro každý obraz této sady obrazů jsem vytvořil soubor pro inicializaci ohraničením. Každý soubor ze sady jsem pak pomocí aplikace segmentoval. Testování probíhalo na procesoru Intel Core i-5 4590 3,7 GHz. V následující tabulce můžete vidět porovnání segmentačních metod aplikace na této sadě obrazů. Nastavení parametrů metod při testování aplikace je uvedeno v podkapitole 5.1.

Metoda	Průměrná chyba	$\sigma$ chyby	Průměrná doba běhu	$\sigma$ doby běhu
Graph Cut	10,8%	9,6%	0,7 s	0,2 s
Grab Cut s ohraničením	15,7%	13,0%	0,4 s	0,4 s
Grab Cut se vzorky	12,8%	10,4%	0,3 s	0,2 s
One Cut s ohraničením	16,8%	13,9%	2,4 s	3,4 s
One Cut se vzorky	11,4%	9,3%	0,3 s	0,2 s

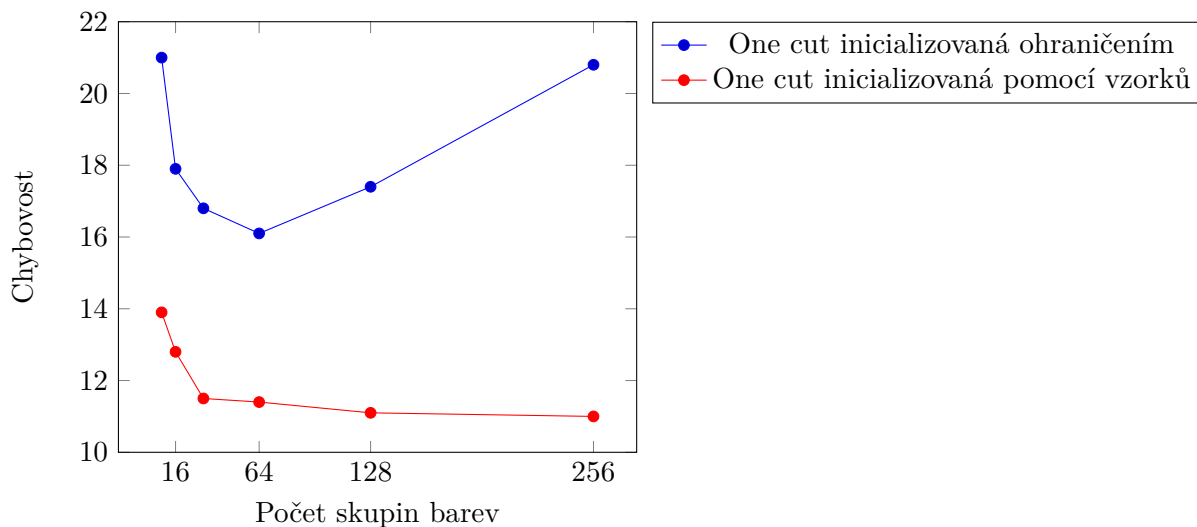
Tabulka 1: Porovnání výsledků segmentačních metod

Průměrná chyba v tabulce určuje kolik procent pixelů bylo zařazeno chybně buďto do pozadí nebo do objektu na základě porovnání se souborem s ideální segmentací a  $\sigma$  udává směrodatnou odchylku. Porovnávání, zda byl daný pixel zařazený správně nebo ne, probíhá pouze u pixelů, které nebyly uživatelem označeny pomocí tvrdých omezení jako pixely patřící buďto do objektu, nebo do pozadí. Z tohoto důvodu je vhodné mezi sebou srovnávat metody, kde je ve vstupním obraze označen objekt a pozadí stejným způsobem, tj. metody inicializované ohraničením s metodami inicializovanými ohraničením a metody inicializované vzorky s metodami inicializovanými vzorky. Průměrná doba běhu pak udává průměrnou dobu běhu dané segmentační metody. Na obrázku 9 je ukázka segmentace, provedené pomocí jednotlivých segmentačních metod.



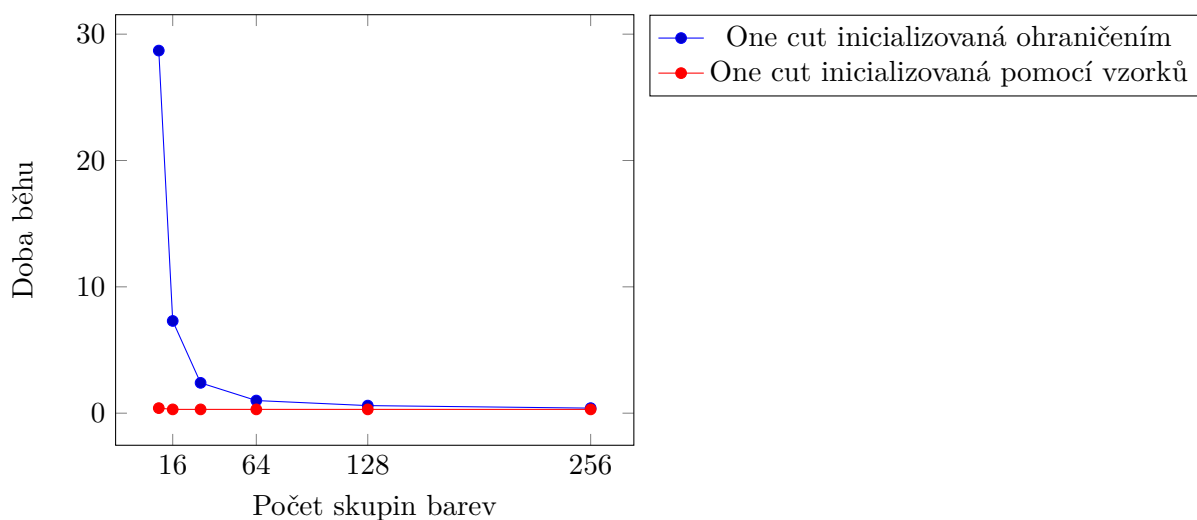
Obrázek 9: Ukázka výsledků segmentace pro (b) Graph cut, (c) Grab cut s ohraňčením, (d) Grab cut se vzorky, (e) One cut s ohraňčením, (f) One cut se vzorky

Z tabulky 1 vyplývá, že metody Graph cut a Grab cut mají v průměru menší chybovost než metoda One Cut inicializovaná ohraňčením. Metoda One cut má však v tabulce 1 nastavený počet skupin barev na  $32^3$ . Při vyšším počtu skupin začne chybovost této metody klesat, jak lze vidět v grafu v obrázku 10. Při počtu skupin barev  $64^3$  dosáhne metoda One cut inicializovaná pomocí ohraňčení nejnížší chybovosti. Pokud je metoda One cut inicializovaná pomocí vzorků, dosáhne nejnížší chybovosti při počtu barevných skupin  $256^3$ . Změna počtu barevných skupin je jedna z možností, jak ovlivnit výsledky segmentační metody. Zvyšováním počtu barevných skupin dochází i ke zvyšování počtu pomocných uzlů v grafu a současně se snižuje množství pixelů, připojených k jednomu pomocnému uzlu. K jednomu pomocnému uzlu jsou pak připojeny jen pixely, které mají velice podobný jas. Díky tomu je pak metoda schopná poskytovat lepší výsledky v případě obrazů, kde je nízký kontrast mezi objektem a pozadím. Poměrně vysoké chybovosti metody Grab cut inicializované pomocí vzorků se budu věnovat v kapitole 5.3.



Obrázek 10: Chybovost metody One cut v závislosti na počtu barevných skupin

Při zvyšování počtu barevných skupin tedy i zvyšování počtu pomocných uzlů v grafu, dochází ke snižování hustoty propojení v grafu a díky tomu se rovněž snižuje i průměrná doba běhu metody One cut, viz graf 11.



Obrázek 11: Doba běhu metody One cut v závislosti na počtu barevných skupin

Metoda One cut s menším počtem skupin barev může poskytovat dobré výsledky při segmentaci obrazů, kde je vysoký kontrast mezi objektem a pozadím. V obrazech, kde je tento kontrast menší a barvy objektu a pozadí jsou si více podobné, má metoda One cut s menším počtem skupin barev větší problém rozlišit objekt a pozadí. Na obrázku 12 je ukázka segmentace pomocí metody One cut s malým počtem barevných skupin na obraze s vysokým kontrastem mezi objektem a pozadím a na obraze s nízkým kontrastem mezi objektem a pozadím. Zvýše-

ním počtu barevných skupin se pak zvýší citlivost metody One cut na barevné rozdíly objektu a pozadí.



Obrázek 12: Ukázka segmentace pomocí metody One cut inicializované ohraňčením s malým počtem barevných skupin na obraze s vysokým kontrastem mezi objektem a pozadím (nahore), a na obraze s nízkým kontrastem mezi objektem a pozadím (dole)

### 5.1 Nastavení parametrů metod při testování aplikace

Výsledky segmentace lze kromě rozmazání obrazu za účelem odstranění šumu, vhodnou volbou ohraňčení a vzorků, nebo vhodným počtem barevných skupin ovlivnit také vhodnou volbou parametrů. U všech metod v aplikaci lze měnit váhu hladkostního členu nastavením konstant  $\lambda$  a  $\sigma$ . Při testování aplikace jsem dospěl k závěru, že aplikace poskytuje nejlepší výsledky v případě, kdy je konstanta  $\sigma$  vypočítána pro každý obraz jako průměrná druhá mocnina Euklidovy vzdálenosti barev sousedících pixelů. Při testování byla  $\lambda$  v jednotlivých metodách pro danou sadu obrazů nastavena stejně, a to na hodnotu 9 u metody One cut a na hodnotu 350 u metody Graph cut.

### 5.2 Porovnání metod inicializovaných pomocí ohraňčení

Jelikož aplikace nabízí pro segmentaci dvě metody inicializované ohraňčením a tři metody inicializované pomocí vzorků, tak je vhodné mezi sebou porovnat stejně inicializované metody. Tato podkapitola bude věnovaná porovnání metod Grab cut a One cut s ohraňčením.

Z tabulky 1 je zřejmé, že na zvolené sadě obrazů dosahuje metoda Grab cut lepších výsledků než metoda One cut, a to co se týče chybovosti i doby běhu. I po zvýšení počtu barevných skupin metody One cut na  $64^3$ , kdy tato metoda dosahuje v průměru nejlepších výsledků, je její průměrná chybovost stále vyšší, než u metody Grab cut.

Jak jsem však zmínil už v úvodu této práce, žádná metoda není ideální a každá metoda se hodí pro určitý typ obrazů. Porovnáváním výsledků segmentace na dané sadě obrazů jsem zjistil, že metoda One cut poskytuje horší výsledky než metoda Grab cut v případě, kdy je hledaný objekt v obraze umístěn tak, že při označení objektu ohraničením pokrývá ohraničení velkou plochu obrazu a současně je i malý kontrast mezi objektem a pozadím. V tomto případě má metoda One cut vyšší chybovost než metoda Grab cut. Tento rozdíl v chybovosti je nejpatrnější, když má metoda One cut nastavený malý počet barevných skupin. Při větším počtu barevných skupin pak začne metoda v tomto případě poskytovat lepší výsledky, ale stále bude mít vyšší chybovost než metoda Grab Cut. Metoda One cut při malém počtu barevných skupin poskytuje oproti metodě Grab cut mnohem lepší výsledky při segmentaci obrazů s vysokým kontrastem mezi objektem a pozadím, jak lze vidět na obrázku 13.



Obrázek 13: Ukázka výsledků segmentace pomocí metody Grab cut (uprostřed) a metody One cut inicializované ohraničením (vpravo) na obrázku s vysokým kontrastem mezi objektem a pozadím

Pokud je naopak použita metoda One cut s velkým počtem skupin barev, poskytuje tato metoda lepší výsledky u obrazů, kde je kontrast mezi objektem a pozadím nízký, viz obrázek 14.



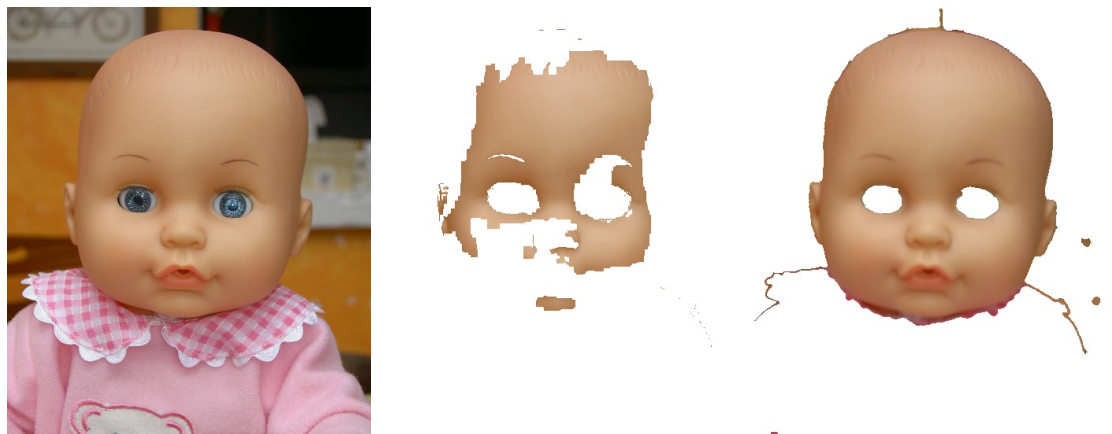
Obrázek 14: Ukázka výsledků segmentace pomocí metody Grab cut (uprostřed) a metody One cut (vpravo) na obrázku s nízkým kontrastem mezi objektem a pozadím

### 5.3 Porovnání metod inicializovaných pomocí vzorků

Dále aplikace nabízí segmentaci metodami, které jsou inicializované pomocí vzorků. Jedná se o metodu Graph cut, metodu One cut inicializovanou pomocí vzorků a metodu Grab cut inicializovanou pomocí vzorků. Tato podkapitola bude věnovaná porovnání výsledků těchto tří metod.



Při pohledu na tabulku 1 je zřejmé, že průměrná chybovost metody Graph cut a metody One cut je téměř stejná, avšak průměrná doba běhu těchto metod se liší. Nejvyšší chybovost má pak metoda Grab cut. Stejně jako v předchozí podkapitole jsem i zde zjišťoval, v kterých případech poskytují jednotlivé metody nejlepší výsledky. Porovnáváním výsledků segmentace obrazů z dané sady jsem došel k závěru, že metoda One cut má menší chybovost než metoda Graph cut u obrazů, kde je nízký kontrast mezi objektem a pozadím a to díky rozdělení barev do skupin, díky čemuž je metoda One cut citlivější na barevné rozdíly objektu a pozadí, jak lze vidět na obrázku 15.



Obrázek 15: Ukázka výsledků segmentace pomocí metody Graph cut (uprostřed) a metody One cut inicializované pomocí vzorků (vpravo) na obrázku s nízkým kontrastem mezi objektem a pozadím

V případě obrazů, kde je kontrast mezi objektem a pozadím vysoký, dokáže metoda Graph cut při správném nastavení konstant, upravujících důležitost hladkostního členu, poskytnout lepší výsledek než metoda One cut inicializovaná pomocí vzorků.

Z tabulky 1 vyplývá, že metoda Grab cut má v porovnání s ostatními metodami, které jsou inicializovány pomocí vzorků, nejvyšší chybovost. Důvodem této vysoké chybovosti je matice, zmíněná v kapitole 4.4.2. V této matici jsou všechny pixely, které nejsou označeny pomocí vzorků jako pixely patřící do objektu nebo do pozadí, označeny jako pixely patřící pravděpodobně do pozadí. Pokud je pak pomocí vzorků označeno malé množství pixelů objektu jako pixely patřící do objektu, má tato metoda tendenci zařadit části objektu do pozadí. Aby tato metoda dosahovala lepších výsledků, musí uživatel označit v objektu dostatečné množství pixelů s různými jasy, viz obrázek 16. Pokud se objekt skládá z několika oblastí z rozdílnými jasy, může být správné označení pomocí vzorků poměrně náročné.

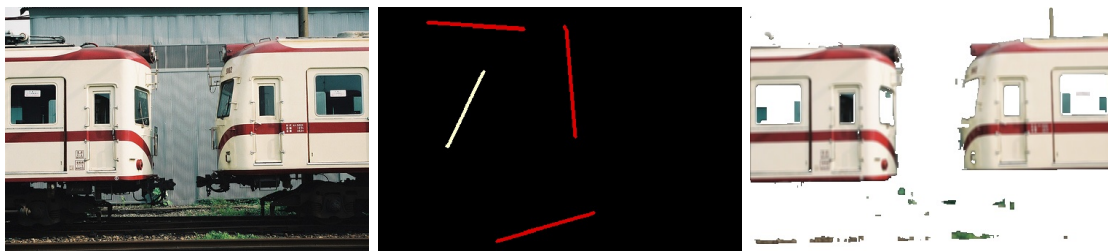


Obrázek 16: Ukázka výsledků segmentace metodou Grab cut pro dvě různá označení pomocí vzorků

#### 5.4 Porovnání metod inicializovaných ohraňčením a metod inicializovaných pomocí vzorků

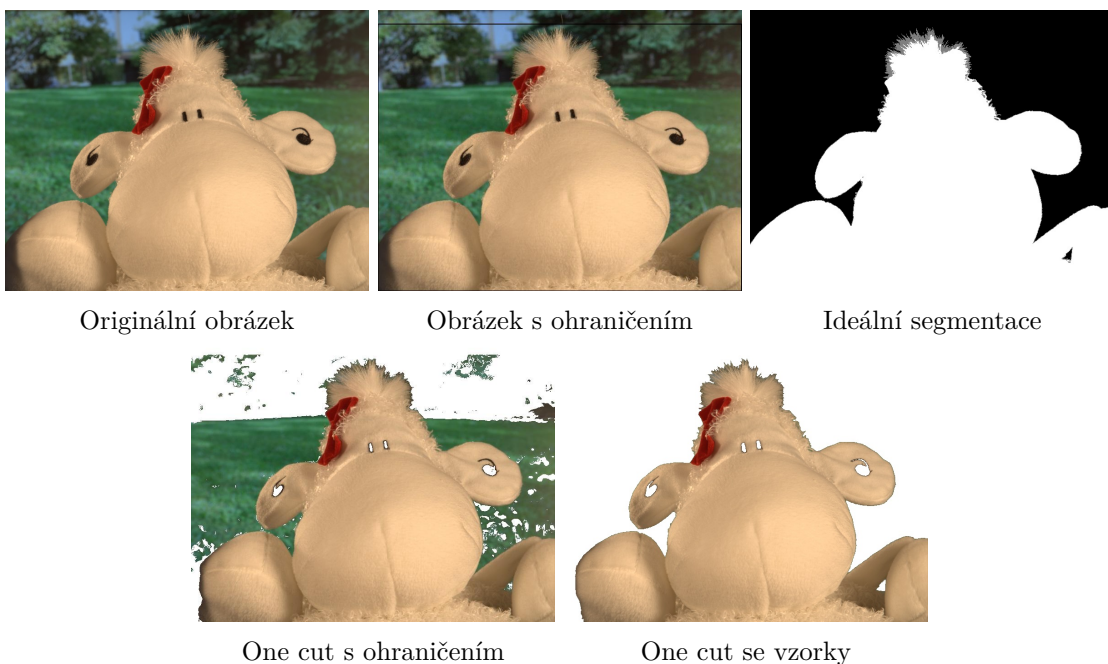
Jak jsem již zmínil v úvodu této kapitoly, nemá smysl porovnávat mezi sebou chybovost metod inicializovaných pomocí vzorků a chybovost metod inicializovaných pomocí ohraňčením. Můžeme však porovnat výhody a nevýhody těchto metod.

Už na základě toho, jak fungují segmentační metody inicializované pomocí vzorků je zřejmé, že tyto metody budou citlivější na vstupní, uživatelem zadávaná data. Pokud například uživatel neoznačí část pozadí, která je podobná objektu jako pozadí, poskytují tyto metody špatné výsledky, viz obrázek 17. Pokud je žádoucí segmentovat jen jeden vagon, je tento výsledek špatný.



Obrázek 17: Ukázka výsledků segmentace s nevhodným označením objektu pomocí vzorků, vlevo je vstupní obraz, uprostřed jsou použité vzorky a vpravo je výsledek segmentace provedené metodou Graph cut

Z fungování metod inicializovaných ohraničením je zase zřejmé, že se nehodí k segmentaci obrazů, v nichž se nachází hledaný objekt vícekrát a chceme-li z obrazu segmentovat všechny tyto objekty, přičemž by ohraničení překrylo velkou část obrazu. Metody inicializované ohraničením mohou také poskytovat horší výsledky v případě, kdy se v ohraničení nachází i část pozadí s určitým jasem, přičemž vně ohraničení se daný jas, nebo jemu podobný, vůbec nenachází, nebo se zde nachází jen v malé míře. V tomto případě může dojít k tomu, že je část pozadí identifikována jako objekt. K uvedenému příkladu může dojít v případě, kdy se objekt dotýká okraje obrazu, viz obrázek 18.



Obrázek 18: Porovnání výsledků segmentace pomocí metody One cut

## 5.5 Testování segmentačních metod na obrazech bez odstranění šumu

Rozmazání obrazu je prvním krokem při segmentaci obrazu a slouží k odstranění šumu. Pro úplnost testování jsem se rozhodl otestovat metody Graph cut, Grab cut a One cut na obrazech z

dané sady, aniž by obrazy byly před segmentací rozmazány. Výsledky pro jednotlivé segmentační metody jsou v tabulce 2.

Metoda	Průměrná chyba	$\sigma$ chyby	Průměrná doba běhu	$\sigma$ doby běhu
Graph Cut	11,5%	10,4%	0,8 s	0,3 s
Grab Cut s ohraničením	16,1%	13,8%	0,5 s	0,5 s
Grab Cut se vzorky	12,0%	10,0%	0,4 s	0,2 s
One Cut s ohraničením	17,5%	15,0%	2,7 s	3,3 s
One Cut se vzorky	11,8%	8,9%	0,3 s	0,3 s

Tabulka 2: Porovnání segmentačních metod na obrazech, které nebyly rozmazány Gaussovým filtrem

Při porovnání tabulky 2 s tabulkou 1 je zřejmé, že průměrná chybovost se po odstranění Gaussova filtru u jednotlivých metod zvýšila jen nepatrně. Jelikož testování probíhalo na obrazech, které jsou v poměrně dobré kvalitě, rozhodl jsem se pro další testování poškodit obrazy z dané sady šumem. Pro vytvoření šumu v obrazech jsem použil Gaussův šum, kde jsem  $\sigma$  nastavil na hodnotu 20 a střední hodnotu jsem nastavil na hodnotu 0. Výsledky pro jednotlivé segmentační metody jsou uvedeny v tabulce 3. Na obrázku 19 je ukázka obrazu znečištěného šumem.



Originální obrázek

Obrázek znečištěný šumem

Obrázek 19: Ukázka obrázku znečištěného šumem

Metoda	Průměrná chyba	$\sigma$ chyby	Průměrná doba běhu	$\sigma$ doby běhu
Graph Cut	12,0%	9,9%	1,2 s	0,6 s
Grab Cut s ohraničením	27,5%	18,3%	0,8 s	1,0 s
Grab Cut se vzorky	12,3%	9,8%	0,6 s	0,9 s
One Cut s ohraničením	26,2%	17,7%	1,3 s	1,4 s
One Cut se vzorky	14,3%	9,5%	0,4 s	0,7 s

Tabulka 3: Porovnání segmentačních metod na obrazech, které nebyly rozmazány Gaussovým filtrem a před zpracováním byly poškozeny šumem

Z porovnání tabulky 3 a tabulky 2 vyplývá, že přidání šumu do obrazů mělo nejmenší vliv na metodu Graph cut a metodu Grab cut inicializovanou pomocí vzorků. Z metod inicializovaných pomocí vzorků, se pak nejvíce zvýšila chybovost metody One cut. U obou metod inicializovaných pomocí ohraničení došlo po přidání šumu ke znatelnému nárůstu chybovosti, přičemž menší vliv mělo přidání šumu na metodu One cut.

## 6 Závěr

Cílem bakalářské práce bylo nastudovat a popsat segmentační metody Graph cut, Grab cut a One cut a naimplementovat aplikaci, provádějící segmentaci obrazu pomocí těchto metod a porovnat jejich výsledky.

Nastudoval jsem si dané segmentační metody a jejich popisu jsem věnoval celou třetí kapitolu, kde jsem je podrobně popsal. Následně jsem naimplementoval aplikaci provádějící segmentaci obrazu pomocí těchto tří metod. Samotná aplikace a implementace jednotlivých segmentačních metod je popsána ve čtvrté kapitole.

Mým posledním úkolem bylo otestovat metody Graph cut, Grab cut a One cut na sadě reálných obrazů, porovnat a zhodnotit dosažené výsledky. Pro otestování metod jsem použil skupinu obrazů, kterou poskytují autoři k článku [22]. Na stejné sadě obrazů jsem pak jednotlivé segmentační metody porovnával. Testování metod a jejich porovnání jsem věnoval pátou kapitolu. Průměrná chybovost jednotlivých metod se pohybovala v rozmezí od 10,9% do 16,8% což vzhledem k tomu, že při měření průměrné chyby měla konstanta  $\lambda$  v jednotlivých metodách po celou dobu stejnou hodnotu, považuji za dobrý výsledek. Testování probíhalo na procesoru Intel Core i-5 4590 3,7 GHz a průměrná doba potřebná pro segmentaci byla 0,87 s. Výpočet by se dal urychlit např. paralelizací částí aplikace, nebo implementací některých částí pro výpočet na grafické kartě.

Díky této aplikaci jsem získal znalosti o segmentaci obrazu, segmentačních metodách založených na grafových řezech, konkrétně metodách Graph cut, Grab cut a One cut a jejich implementaci. Při testování aplikace jsem došel k výsledku, že každá z těchto metod poskytuje nejlepší výsledky pro určitou skupinu obrazů. Možným rozšířením aplikace by mohla být například již zmiňovaná paralelizace.

## Literatura

- [1] M. Tang, L. Gorelick, O. Veksler, and Y. Boykov, “Grabcut in one cut,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1769–1776, 2013.
- [2] Y. Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in nd images,” in *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, vol. 1, pp. 105–112, IEEE, 2001.
- [3] C. Rother, V. Kolmogorov, and A. Blake, “Grabcut: Interactive foreground extraction using iterated graph cuts,” in *ACM transactions on graphics (TOG)*, vol. 23, pp. 309–314, ACM, 2004.
- [4] I. M. Španěl, “Vysoké učení technické v brně.” <http://www.fit.vutbr.cz/~spanel/segmentace/>. [Citováno 26.3.2019].
- [5] M. S. University, “Biometrics research group.” <http://biometrics.cse.msu.edu/info/index.html>, 2009.
- [6] R. C. da Silva Vasconcelos and H. Pedrini, “Fingerprint image segmentation based on oriented pattern analysis,” in *14th International Conference on Computer Vision Theory and Applications*, pp. 405–412, 2019.
- [7] P. Oad and W. Ahmad, “Iris localization using daugman’s algorithm,” 2012.
- [8] J. Almotiri, K. Elleithy, and A. Elleithy, “Retinal vessels segmentation techniques and algorithms: A survey,” *Applied Sciences*, vol. 8, 01 2018.
- [9] A. Norouzi, M. S. M. Rahim, A. Altameem, T. Saba, A. E. Rad, A. Rehman, and M. Uddin, “Medical image segmentation methods, algorithms, and applications,” *IETE Technical Review*, vol. 31, no. 3, pp. 199–213, 2014.
- [10] W. contributors, “Volume rendering — Wikipedia, the free encyclopedia.” [https://en.wikipedia.org/wiki/Volume\\_rendering](https://en.wikipedia.org/wiki/Volume_rendering), 04 2020. [Citováno 20.4.2020].
- [11] <https://www.rsipvision.com/adas/>. [Citováno 15.2.2019].
- [12] U. of Cambridge, “Segnet.” <https://mi.eng.cam.ac.uk/projects/segnet/>. [Citováno 4.2.2020].
- [13] P. Mgr. Martin Pilát, “Neuronové sítě - konvoluční sítě a zpracování obrazu.” <https://martinpilat.com/cs/prirodou-inspirovane-algoritmy/neuronove-site-konvolucni-site-zpracovani-obrazu>. [Citováno 4.2.2020].
- [14] “Model targets.” <https://library.vuforia.com/features/objects/model-targets.html>.

- [15] “Eon reality creates next generation augmented reality solution with sats for airport ramp handling services.” <https://eonreality.com/eon-reality-creates-next-generation-augmented-reality-solution-sats-airport-ramp-handling-services/>. [Citováno 1.4.2020].
- [16] <https://www.bioss.ac.uk/people/chris/ch4.pdf>. [Citováno 26.3.2019].
- [17] M. M. Viktor Haškovec, “Segmentace obrazu s využitím hysterezního prahování,” *Vysoká škola chemicko-technologická v Praze, Ústav počítačové a řídicí techniky*, 2004.
- [18] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proc. 8th Int’l Conf. Computer Vision*, vol. 2, pp. 416–423, July 2001.
- [19] E. Sojka, *Digitální zpracování a analýza obrazu*. Vysoká Škola báňská-Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2000.
- [20] M. Kaur and P. Goyal, “A review on region based segmentation,” *International Journal of Science and Research (IJSR)*, vol. 4, no. 4, pp. 3194–3197, 2015.
- [21] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 9, pp. 1124–1137, 2004.
- [22] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman, “Geodesic star convexity for interactive image segmentation,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3129–3136, IEEE, 2010.